

LDAP challenge

Michael Kluge

June 4, 2013

Table of contents

- 1 LDAP setup
 - Installation
 - Configuration
 - First test
- 2 User administration with LDAP
 - Some benefits
 - Which classes does LDAP provide for storing users and groups?
 - How to store users and groups using LDAP?
 - How to use LDAP as nameservice?
- 3 Access control statements in LDAP
- 4 Enable TLS encryption
- 5 Setup master-slave replication
- 6 phpLDAPadmin: a web-based LDAP client

Meaning of different box colors

`/path/to/some/file`

Content of the file or just parts of it.

Shell commands

Can be executed on the shell.

`ldapvi`

LDAP database opened with `ldapvi`. (`dc=tbl` or `cn=config`)

LDAP setup

Package installation

install with apt-get

```
apt-get install slapd
```

```
apt-get install ldap-utils
```

```
apt-get install ldapvi
```

- ▶ slapd (LDAP daemon provided by OpenLDAP)
- ▶ ldap-utils (contains ldapadd, ldapmodify, ldapsearch and more)
- ▶ ldapvi (LDAP client which allows to edit entries comfortable)

Configuration of slapd

reconfigure slapd

dpkg-reconfigure slapd

- ▶ set base dn to tbl
- ▶ enter some organization name
- ▶ set admin password
- ▶ use default storage format HDB
- ▶ remove database, if slapd is purged
- ▶ move old database to /var/backups
- ▶ only allow new LDAP version 3

Make ldapvi use vim

- ▶ ldapvi may use nano as default editor
- ▶ set your preferred text editor using environment variables **EDITOR** and **VISUAL**
- ▶ make it permanent by adding it to your `~/.bashrc` file

set environment variables

```
export EDITOR=vim;  
export VISUAL=$EDITOR;
```

Own logfile (default: /var/log/syslog)

add to /etc/rsyslog.conf

```
local4.*    /var/log/sldap.log
```

create /etc/logrotate.d/slapd

```
/var/log/slapd.log
{
    rotate 7
    daily
    missingok
}
```

restart service

```
/etc/init.d/rsyslog restart
```


Check, if ldap daemon is working

open database with ldapvi

```
ldapvi -h ldapi:/// -discover -D cn=admin,dc=tbl
```

dc=tbl

0 dc=tbl

objectClass: top

objectClass: dcObject

objectClass: organization

o: TUM

dc: tbl

1 cn=admin,dc=tbl

objectClass: simpleSecurityObject

objectClass: organizationalRole

cn: admin

description: LDAP administrator

userPassword: {SSHA}EDxDhVgSgxxxxxxxxxxxx8HvIY5IzYI1

User administration with LDAP

Some benefits

- ▶ better to maintain in case of many users and groups
- ▶ centralized solution

Which classes does LDAP provide for storing users and groups?

class	mandatory	optional
posixAccount		
shadowAccount		
inetOrgPerson		
posixGroup		

Which classes does LDAP provide for storing users and groups?

class	mandatory	optional
posixAccount	cn, uid, uidNumber, gidNumber, homeDirectory	
shadowAccount	uid	
inetOrgPerson	extends organizationalPerson	
posixGroup	cn, gidNumber	

Which classes does LDAP provide for storing users and groups?

class	mandatory	optional
posixAccount	cn, uid, uidNumber, gidNumber, homeDirectory	userPassword, loginShell, gecos, description
shadowAccount	uid	userPassword, shadowLastChange, shadowMin, shadowMax, shadowWarning, shadowInactive, shadowExpire, shadowFlag, description
inetOrgPerson	extends organizationalPerson	f.e. sn, mail, preferredLanguage, roomNumber, employeeType
posixGroup	cn, gidNumber	userPassword, memberUid, description

How to get these information?

- ▶ RFC = Request for Comments
- ▶ syntax of **olcObjectClasses** defined in RFC 2252
- ▶ **posixGroup** defined in RFC 2307
- ▶ `cn=schema` holds all hard-coded schema definitions of slapd

`cn=schema,cn=config`

```
olcObjectClasses: {2}( 1.3.6.1.1.1.2.2 NAME 'posixGroup' DESC  
'Abstraction of a group of accounts' SUP top STRUCTURAL MUST ( cn  
$ gidNumber ) MAY ( userPassword $ memberUid $ description ) )
```

Where to get information about user and groups?

example out of /etc/group

```
michael:x:1009:
```

example out of /etc/passwd

```
michael:x:1009:1009:Michael Kluge,,,:/home/michael:/bin/bash  
git:x:1066:1066:,,,:/home/git:/usr/bin/git-shell
```

example out of /etc/shadow

```
michael:XfICq3PDN.....y9ISfjHOEG1:15827:0:99999:7:::
```

generation of password hash

```
slappasswd -s password > /tmp/testpasswd
```

```
/tmp/testpasswd
```

```
{SSHA}3MCvXDfkaeC2bMkD5JWzhwRtgBCRwNvD
```


How to store groups in LDAP syntax?

open database with ldapvi

```
ldapvi -h ldapi:/// -discover -D cn=admin,dc=tbl
```

group in LDAP syntax

```
add ou=group,dc=tbl  
objectClass: organizationalUnit  
ou: group
```

```
add cn=michael,ou=group,dc=tbl  
objectClass: posixGroup  
cn: michael  
gidNumber: 1009
```

- ▶ add these lines to the file
- ▶ save file and type 'y'
- ▶ ldapvi will apply the changes to the database

How to store users in LDAP syntax?

user in LDAP syntax

```
add ou=people,dc=tbl
objectClass: organizationalUnit
ou: people
```

```
add uid=michael,ou=people,dc=tbl
objectClass: posixAccount
objectClass: shadowAccount
objectClass: inetOrgPerson
cn: michael
uid: michael
uidNumber: 1009
gidNumber: 1009
homeDirectory: /home/michael
loginShell: /bin/bash
description: LDAP posixUser!!!
userPassword: {CRYPT}XfICq3PDN.....y9ISfjHOEG1
sn: Kluge
preferredLanguage: de, en
```

Package installation

install with apt-get

```
apt-get install libnss-ldap
```

- ▶ enter **cn=admin,dc=tbl** when asked for LDAP admin account

install with apt-get

```
apt-get install libpam-ldap
```

```
apt-get install nscd
```

- ▶ libnss-ldap (**N**ame **S**ervice **S**witch: module for using LDAP as name service)
- ▶ libpam-ldap (interface between PAM authentication system and LDAP)
- ▶ nscd (**N**ame **S**ervice **C**ache **D**aemon: caches name service data)

Configuration of libnss-ldap

```
edit /etc/nsswitch.conf
```

```
passwd:    compat ldap
```

```
group:     compat ldap
```

```
shadow:    compat ldap
```

```
configure PAM
```

```
pam-auth-update
```

- ▶ enable **LDAP Authentication**

Test it!

configure PAM

```
getent passwd | grep " michael" > /tmp/users
```

/tmp/users

```
michael:x:1009:1009:Michael Kluge,,,:/home/michael:/bin/bash  
michael:x:1009:1009:michael:/home/michael:/bin/bash
```

- ▶ first entry comes from /etc/passwd
- ▶ second entry comes from LDAP
- ▶ same possible with getent for group, shadow and more
- ▶ comment out user/group from /etc/[passwd|shadow|group]

invalidate nscd cache or stop daemon

```
nscd -i passwd  
nscd -i group
```

- ▶ try to login with user

Access control statements in LDAP

Syntax in LDAP

syntax for access control

access to <what> [by <who> [<access>] [<control>]]+

Syntax in LDAP

syntax for access control

access to <what> [by <who> [<access>] [<control>]]+

- ▶ <what> : entity the access control directive applies to
 - ▶ f.e. dn=<dnpattern> → dn=tbl

Syntax in LDAP

syntax for access control

access to <what> [by <who> [<access>] [<control>]]+

- ▶ <what> : entity the access control directive applies to
 - ▶ f.e. dn=<dnpattern> → dn=tbl
- ▶ <who> : to whom the access rules apply to
 - ▶ multiple blocks are allowed
 - ▶ is terminated by 'by * none stop'
 - ▶ f.e anonymous, users, group, self, regex, dn, * and more

Syntax in LDAP

syntax for access control

access to <what> [by <who> [<access>] [<control>]] +

- ▶ <what> : entity the access control directive applies to
 - ▶ f.e. dn=<dnpattern> → dn=tbl
- ▶ <who> : to whom the access rules apply to
 - ▶ multiple blocks are allowed
 - ▶ is terminated by 'by * none stop'
 - ▶ f.e. anonymous, users, group, self, regex, dn, * and more
- ▶ <access> : specific access privileges to <who>
 - ▶ <level> or <priv>
 - ▶ f.e. none, read, write, manage (<level>)
 - ▶ f.e. =0, +r, +w, +m, -z (<priv>)

Syntax in LDAP

syntax for access control

access to <what> [by <who> [<access>] [<control>]]+

- ▶ <what> : entity the access control directive applies to
 - ▶ f.e. dn=<dnpattern> → dn=tbl
- ▶ <who> : to whom the access rules apply to
 - ▶ multiple blocks are allowed
 - ▶ is terminated by 'by * none stop'
 - ▶ f.e. anonymous, users, group, self, regex, dn, * and more
- ▶ <access> : specific access privileges to <who>
 - ▶ <level> or <priv>
 - ▶ f.e. none, read, write, manage (<level>)
 - ▶ f.e. =0, +r, +w, +m, -z (<priv>)
- ▶ <control> : controls the flow of access rule application
 - ▶ stop: default; access checking stops in case of match
 - ▶ continue: other <who> may in the same <access> may alter rights
 - ▶ break: other <access> which match the same target are processed

Examples from cn=config

example 1

```
olcAccess: to *  
by dn.exact=gidNumber=0+uidNumber=0,  
           cn=peercred,cn=external,cn=auth manage  
by * break
```

- ▶ full access on everything to root
- ▶ maybe other <access> blocks are applied

Examples from cn=config

example II

```
olcAccess: to *  
by self write by dn="cn=admin,dc=tbl" write  
by * read
```

- ▶ write access on every own entry to users
- ▶ write access on everything to admin
- ▶ all other users have only read access

Examples from cn=config

example III

```
olcAccess: to attrs=userPassword,shadowLastChange
by self write
by anonymous auth
by dn="cn=admin,dc=tbl" write
by * none
```

- ▶ **write** access to **users** on their own **passwords**
- ▶ **anonymous users** may use **passwords** for **authentication/authorization**
- ▶ **write** access to **admin** on all **passwords**
- ▶ **all other users** have **no rights**

Enable TLS encryption

Certificate generation

- ▶ use tinyca2 from tinyca to generate new certificate authority (CA)
- ▶ generate key and certificate for server
- ▶ use hostname of the server as common name
- ▶ export certificates as pem file
- ▶ remove passphrase from private key

TinyCA2 Management 0.7.5

CA Preferences Help

Create CA

Create a new CA

Name (for local storage): michael.tbl

Data for CA Certificate

Common Name (for the CA): michael.tbl

Country Name (2 letter code): DE

Password (needed for signing): ●●●

Password (confirmation): ●●●

State or Province Name:

Locality Name (eg. city):

Organization Name (eg. company): TUM

Organizational Unit Name (eg. section):

eMail Address: michael.kluge@campus.lmu.de

Valid for (Days): 99999

Keylength: ☐ 1024 ☐ 2048 ☒ 4096

Digest: ☒ SHA-1 ☐ MD2 ☐ MDC2 ☐ MD4 ☐ MD5 ☐ RIPEMD-160

OK Cancel

Tiny CA Management 0.7.5 - michael.tbl

Preferences Help

Certificates Keys Requests

Create Request

Create a new Certificate Request

Common Name (eg. your Name, your eMail Address or the Servers Name): michael.tbl

eMail Address:

Password (protect your private Key): ●●●●●●

Password (confirmation): ●●●●●●

Country Name (2 letter code): DE

State or Province Name:

Locality Name (eg. city):

Organization Name (eg. company): TUM

Organizational Unit Name (eg. section):

Keylength: ☒ 4096 ☐ 1024 ☐ 2048

Digest: ☒ SHA-1 ☐ MD2 ☐ MDC2 ☐ MD4 ☐ MD5 ☐ RIPEMD-160

Algorithm: ☒ RSA ☐ DSA

OK Cancel

Configuration of LDAP daemon

move certificates to /etc/ldap/ssl

```
mkdir /etc/ldap/ssl  
mv /path/2/cacert.pem /etc/ldap/ssl/cacert.pem  
mv /path/2/ldapcert.pem /etc/ldap/ssl/ldapcert.pem  
mv /path/2/ldapkey.pem /etc/ldap/ssl/ldapkey.pem  
chown -R openldap /etc/ldap/ssl  
chmod -R go-rwx,u+rw /etc/ldap/ssl
```

add to cn=config

```
olcTLSCACertificateFile: /etc/ldap/ssl/cacert.pem  
olcTLSCertificateFile: /etc/ldap/ssl/ldapcert.pem  
olcTLSCertificateKeyFile: /etc/ldap/ssl/ldapkey.pem  
olcTLSVerifyClient: allow  
olcServerID: 9
```

restart service

```
/etc/init.d/sldap restart
```

Configuration of LDAP client

move CA certificate to /etc/ldap/ssl

```
mkdir /etc/ldap/ssl
```

```
mv /path/2/cacert.pem /etc/ldap/ssl/cacert.pem
```

```
chmod -R 644 /etc/ldap/ssl
```

add to /etc/ldap/ldap.conf

```
ldap_version 3
```

```
ssl start_tls
```

```
TLS_REQCERT demand
```

```
TLS_CACERT /etc/ldap/ssl/cacert.pem
```

test it

```
ldapsearch -D cn=admin,dc=tbl -b dc=tbl -h localhost -ZZ -W
```

/var/log/sldap.log

```
daemon: listen=8, new connection on 17
```

```
conn=1003 fd=17 ACCEPT from IP=127.0.0.1:56393 (IP=0.0.0.0:389)
```

```
conn=1003 fd=17 TLS established tls_ssf=128 ssf=128
```

Setup master-slave replication

Configure master (provider) I

- ▶ modules can not be configured while daemon is running
- ▶ stop daemon, make changes and start it again

stop service

```
/etc/init.d/sldap stop
```

add to `/etc/ldap/slapd.d/cn=config/cn=module{0}.ldif`

```
olcModuleLoad: {1}syncprov
```

start service

```
/etc/init.d/sldap start
```

Configure master (provider) II

cn=config

```
add olcOverlay={0}syncprov,olcDatabase={1}hdb,cn=config
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
olcOverlay: {0}syncprov
olcSpCheckpoint: 100 10
olcSpSessionlog: 100
olcSpNoPresent: TRUE
olcSpReloadHint: TRUE
```

- ▶ overlays extend or change database operations
- ▶ olcSpCheckpoint: controls when data is written from memory to disk
- ▶ olcSpSessionlog: number of write operations in log
- ▶ olcSpNoPresent: should be set, if session log is used
- ▶ olcSpReloadHint: allows client to force the complete transfer of the database

Configure slave (consumer)

- ▶ install and configure ldap daemon in your L2 as you did before
- ▶ add syncprov module
- ▶ set correct olcServerID in cn=config

```
olcDatabase={1}hdb,cn=config
```

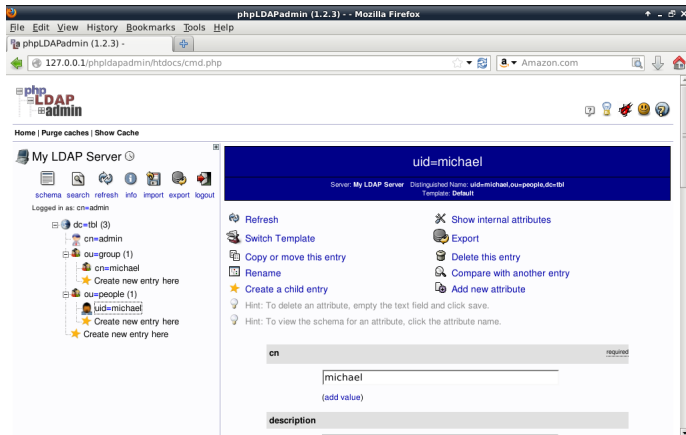
```
olcSyncrepl: {0}rid=123 provider="ldap://michael.tbl" type=refreshOnly  
interval=00:00:00:15 searchbase="dc=tbl" bindmethod=simple  
binddn="cn=admin,dc=tbl" credentials=password4user  
olcUpdateRef: ldap://michael.tbl
```

- ▶ restart daemon
- ▶ test it

phpLDAPAdmin: a web-based LDAP client

phpLDAPadmin

- ▶ a webserver with PHP support is required
- ▶ ldap module for PHP must be installed (not included per default)



Thank you for your attention!